REMARKS

In view of the following discussion, the Applicants submit that none of the claims now pending in the application is made obvious under the provisions of 35 U.S.C. §103. Thus, the Applicants believe that all of these claims are now in allowable form.

I. REJECTION OF CLAIMS 1-42 UNDER 35 U.S.C. § 103

1. Claims 1-7, 10-17, 20-29 and 31-42

Claims 1-7, 10-17, 20-29 and 31-42 stand rejected as being obvious over the Swildens et al. patent (U.S. Patent No. 6,694,358, issued February 17, 2004, hereinafter "Swildens") in view of the Peterson patent (U.S. Patent No. 6,788,648, issued September 7, 2004, hereinafter "Peterson") in further view of Nepustil (U.S. Patent No. 6,240,454, issued May 29, 2001, hereinafter "Nepustil"). The Applicants respectfully traverse the rejection.

Swildens teaches a performance computer network method. Specifically, Swildens teaches a load balancing method that determines the traffic loads (*e.g.*, volume of processing requests) on a plurality of web servers. These various traffic loads are then compared to identify the web server that has the smallest traffic load among the plurality of web servers, and traffic is directed to this server.

Peterson teaches a method and apparatus for load balancing a distributed processing system. Specifically, Peterson teaches a load monitor that monitors the statuses of various processing nodes (*e.g.*, servers). Each processing node sends updates to the load monitor including information about the processing node's current load status. The load monitor compares a processing node's update against a threshold for that processing node, and, if the processing node's current load exceeds a threshold for that processing node, the load monitor takes the processing node offline (*e.g.*, makes the processing node unavailable for further processing requests) until the load falls below the threshold.

Nepustil teaches a dynamic reconfiguration of network servers. The invention comprises a plurality of servers for processing client requests, wherein at least one first server of the plurality of servers has first information and second information related to

the first information, for processing portions of the client requests that require the first information and portions of the client requests that require the second information. (See Nepustil, col. 2, ll. 20-46.) If processing on the at least one server becomes excessive, then the at least one server processes the portions of the client requests which require the first information without also processing the portions of the client requests which require the second information. (See *Id.*) The portions of the client request which require the second information is redirected to at least one second server for processing. (See *Id.*)

The Examiner's attention is directed to the fact that Swildens, Peterson and Nepustil, singly and in any permissible combination, fail to teach, show or suggest determining a load on a primary server and offloading a processing request to another server only if a processing threshold is exceeded at the primary server, as positively claimed by the Applicants. Specifically, Applicants' independent claims 1, 11, 21, 22, 23, 32, 41 and 42 recite:

> 1. A method, in a network comprising a primary server and at least one offload server, for dynamic offloading of processing requests from said primary server to said at least one offload server, the method comprising the steps of:
> determining a load on said primary server;
> if the load on said primary server is less than a first threshold, serving processing requests at said primary server; and
> only if the load on said primary server exceeds said first threshold, then offloading at least a portion of said processing requests to said at least one offload server. (Emphasis added)

> 11. A network apparatus comprising a primary server and at least one offload server connected by an IP-based network, for dynamic offloading of processing requests from said primary server to said at least one offload server, the network apparatus comprising:
> a load controller connected between said network and said primary server;
> a memory connected to said load controller and including data and control instructions for operating said primary server to perform the steps of:
> determining a load on said primary server;
> if said load on said primary server is less than a first threshold, serving processing requests at said primary server; and
> only if said load on said primary server exceeds said first threshold, then offloading at least a portion of said processing requests to said at

least one offload server. (Emphasis added)

21. A system, including an IP network comprising a primary server and at least one offload server, for dynamic offloading of processing requests from said primary server to said at least one offload server, the system comprising:

means for determining a load on said primary server;

means for, if said load on said primary server is less than a first threshold, serving the processing requests at said primary server; and

means for, only if said load on said primary server exceeds said first threshold, offloading at least a portion of said processing requests to said at least one offload server. (Emphasis added)

22. A program product including control instructions for controlling the operation of a computer, said program product operative with said control instructions to operate said computer in an IP-based network comprising a primary server and at least one offload server to dynamically offload processing requests from said primary server to said at least one offload server, the computer operative with said control instructions to perform the steps of:

determining a load on said primary server;

if said load on said primary server is less than a first threshold, serving the processing requests at said primary server; and

only if said load on said primary server exceeds said first threshold, then offloading at least a portion of said processing requests to said at least one offload server. (Emphasis added)

23. A system for allocating processing requirements on a network between a primary server and an offload server, comprising:

a load controller connected to said network for receiving processing requests from clients on said network and allocating said processing requests between said primary and offload servers;

a memory connected to said load controller and storing threshold data and control software for analyzing said threshold data and operating said load controller;

said load controller operative with the threshold data and control software to perform the steps of:

periodically evaluating said processing requests to determine a load on said primary server;

if said load exceeds a first threshold, for a predetermined period of time directing at least one processing request to said offload server; and

only if said load does not exceed said first threshold, directing said processing requests to said primary server. (Emphasis added)

32. A method for allocating processing requirements on an IP network between a primary server and an offload server, comprising:

periodically evaluating processing requests to determine a load on said primary server;

if said load exceeds a first threshold, for a predetermined period of time directing at least one processing request to said offload server; and

only if said processing load does not exceed said first threshold, directing said processing requests to said primary server. (Emphasis added)

41. A system for allocating processing requirements on an IP network between a primary server and an offload server, comprising:

means for periodically evaluating processing requests to determine a load on said primary server;

means for, if said load exceeds a first threshold, for a predetermined period of time directing at least one processing request to said offload server; and

means for, only if said processing load does not exceed said first threshold, directing said processing requests to said primary server. (Emphasis added)

42. A program product including control instructions for controlling the operation of a computer, said program product operative with said control instructions to operate said computer in an IP-based network comprising a primary server and at least one offload server to dynamically offload processing requests from said primary server to said at least one offload server, the computer operative with said control instructions to perform the steps of:

periodically evaluating processing requests to determine a load on said primary server;

if said load exceeds a first threshold, for a predetermined period of time directing at least one processing request to said offload server; and

only if said load does not exceed said first threshold, directing said processing requests to said primary server. (Emphasis added)

Applicants' invention is directed to a system and method for dynamically allocating processing on a network amongst multiple network servers. As the Internet continues to grow, so does traffic (*e.g.*, processing requests) directed to popular Internet servers on the World Wide Web. To support these high traffic rates, many techniques have been developed that use offload servers associated with primary web servers to

process some of the traffic targeted to primary web servers. In many conventional cases, traffic is directed to these offload servers regardless of the current resource availability at the primary web servers. However, it is normally only during peak traffic periods that the offload servers are actually needed; during non-peak periods, substantial amounts of processing resources at the primary server may go unused due to the use of the offload servers. Thus, resources are wasted by using offload servers when they are not needed.

Applicants' invention provides a method for dynamically offloading traffic from a primary server to an offload server or servers based on the current load at the primary server and one or more thresholds. In one particular embodiment, the load at the primary server is first determined. If the load falls below a first threshold (*e.g.*, the primary server is not currently over-loaded), then the traffic is directed to the primary server. However, if the load at the primary server exceeds the first threshold (*e.g.*, the primary server is over-loaded or is currently processing the maximum desirable amount of traffic), then at least a portion of the traffic is directed to an offload server. The first threshold may be based on a number of parameters, including network load, CPU utilization, connections per second, various bandwidth loads, various memory loads and the like. In this way, a web site can make use of excess processing capacity, offloading traffic only when the traffic exceeds the web site's processing capacity. Thus, the Applicants' invention provides cost savings by drastically reducing offloaded work.

In contrast, Swildens, Peterson and Nepustil, singly and in any permissible combination, fail to teach, show or suggest determining a load on a primary server and offloading a processing request to another server only if a processing threshold is exceeded at the primary server. Particularly, neither Swildens nor Peterson teaches a preference for the use of a specified or primary server (*e.g.*, to reduce offloading costs). Rather Swildens and Peterson both require the monitoring of a plurality of servers that may potentially be used to process traffic. Thus, both Swildens and Peterson clearly teach away from the Applicants' invention because both Swildens and Peterson require more data collection than the Applicants' invention does. Applicants' invention determines the load on only one server, unlike Swildens and Peterson, which both

determine the load on a plurality of servers. Because it monitors fewer servers, the Applicants' invention requires less calculations and less comparisons and is less time consuming than determining the load traffic data at a plurality of servers, comparing the loads of each server to the loads of other servers and/or respective thresholds and determining which web server to use for processing from among the plurality.

Moreover, Nepustil fails to bridge the substantial gap left by Swildens and Peterson. Nepustil also fails to teach, show or suggest determining a load on a primary server and offloading a processing request to another server only if a processing threshold is exceeded at the primary server. Particularly, Nepustil also fails to teach a preference for the use of a specified or primary server (*e.g.*, to reduce offloading costs). In fact, Nepustil teaches away from the Applicants' invention because Nepustil specifically teaches that each server checks if its present processing load exceeds a load limit. Specifically, Nepustil states:

> "As is conventional, each server 105-107 keeps a record of its present processing load, for example, in the form of a number of accesses (requests) served per unit of time." (See Nepustil, Col. 4, lines 28-31; Col. 5, lines 14-17 and lines 34-37.)

Consequently, Nepustil has the exact same limitation as taught by Swildens and Peterson which requires the monitoring of a plurality of servers, thereby resulting in more calculations than the Applicants' invention. Therefore, Applicants respectfully submit that independent claims 1, 11, 21, 22, 23, 32, 41 and 42 are clearly patentable and not made obvious by Swildens in view of Peterson, and further in view of Nepustil.

Furthermore, dependent claims 2-7, 10, 12-17, 20, 24-29, 31, and 33-40 depend, either directly or indirectly, from claims 1, 11, 21, 22, 23, and 32 and recite additional limitations. As such, and for at least the exact same reason set forth above, the Applicants submit that claims 2-7, 10, 12-17, 20, 24-29, 31, and 33-40 are also patentable and not made obvious by Swildens in view of Peterson and further in view of Nepustil. As such, the Applicants respectfully request the rejection of claims 1-7, 10-17, 20-29 and 31-42 under 35 U.S.C. § 103 be withdrawn.

2. Claims 8-9, 18-19 and 30

Claims 8-9, 18-19 and 30 stand rejected as being obvious over Swildens in view of Peterson and Nepustil and further in view of the Gupta et al. patent (U.S. Patent No. 6,374,305, issued April 16, 2002, hereinafter "Gupta"). The Applicants respectfully traverse the rejection.

Swildens, Peterson and Nepustil have been discussed above. Gupta teaches a web applications interface system in a mobile-based client-server system. Specifically, Gupta teaches architecture that incorporates two specialized software layers: a specialized "proxy" layer that resides on a mobile client station, and a "web agent" that resides on a server. These layers employ respective memory caches and intelligent filtering capabilities, thereby reducing redundant or otherwise unwanted message transmission.

As discussed above, Swildens, Peterson and Nepustil fail to teach, show or suggest determining a load on a primary server and offloading a processing request to another server only if a processing threshold is exceeded at the primary server, as positively claimed by the Applicants' independent claims 1, 11 and 23. Gupta fails to bridge this gap in the teachings of Swildens, Peterson and Nepustil. Thus, claims 1, 11 and 23 are not made obvious by Swildens in view of Peterson and Nepustil and further in view of Gupta.

Dependent claims 8-9, 18-19 and 30 depend, either directly or indirectly, from claims 1, 11 and 23 and recite additional limitations. As such, and for at least the exact same reasons set forth above, the Applicants submit that claims 8-9, 18-19 and 30 are also not made obvious by the teachings of Swildens in view of Peterson and Nepustil and further in view of Gupta.

II. VOLUNTARY AMENDMENTS

The Applicants have voluntarily amended claims 31 and 40 in order to correct minor typographical errors. Specifically, claims 31 and 40 have been amended to correct the antecedent basis of "at least a portion of one processing request" to "said at least one processing request". The Applicants submit that the amendments to claims

31 and 40 were not made in view of the cited prior art.

III. CHANGE OF CORRESPONDENCE ADDRESS

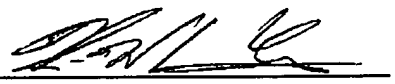
The Applicants again resubmit an Authorization to Act in a Representative Capacity that was previously filed on May 4, 2005. It should be noted that the Applicants have also requested a change of Correspondence Address in the Authorization to Act in a Representative Capacity. It is respectfully requested that the USPTO acknowledges this change.

IV. CONCLUSION

Thus, the Applicants submit that all of the presented claims fully satisfy the requirements of 35 U.S.C. §103. Consequently, the Applicants believe that all of the presented claims are presently in condition for allowance. Accordingly, both reconsideration of this application and its swift passage to issue are earnestly solicited.

If, however, the Examiner believes that there are any unresolved issues requiring the maintenance of the present final action in any of the claims now pending in the application, it is requested that the Examiner telephone Mr. Kin-Wah Tong, Esq. at (732) 530-9404 so that appropriate arrangements can be made for resolving such issues as expeditiously as possible.

Respectfully submitted,



February 27, 2006
Date

Kin-Wah Tong, Attorney
Reg. No. 39,400
(732) 530-9404

Patterson & Sheridan, LLP
595 Shrewsbury Avenue
Shrewsbury, New Jersey 07702